

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-213129

(43)Date of publication of application : 04.08.1992

(51)Int.Cl.

G06F 12/02

(21)Application number : 03-031890

(71)Applicant : HEWLETT PACKARD CO <HP>

(22)Date of filing : 31.01.1991

(72)Inventor : GARCIA PHILIP

(30)Priority

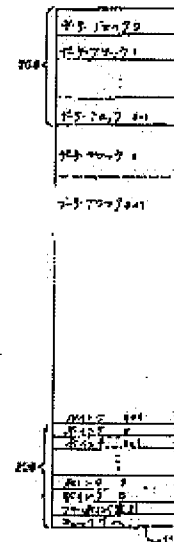
Priority number : 90 472055 Priority date : 31.01.1990 Priority country : US

(54) SYSTEM AND METHOD FOR MEMORY MANAGEMENT

(57)Abstract:

PURPOSE: To improve the efficiency of data storage to a data memory and make its retrieval speed fast by starting growing a pointer table from one end of the data memory and a data field from the other end.

CONSTITUTION: The allocation of the pointer table 206 is started at one end of a data storage area 116. The pointer table 206 contains a painter indicating a data block in the data field 204. Data blocks begin to be allocated from the other end of the data storage area 116 in a pointer table 206. The data field 204 is a variable length part in the data storage area 116 and stores the said data blocks. As a new item is added, the data field 204 and pointer table 206 increase mutually. Consequently, the memory is allocated with the efficiency of a linked list system and data are retrieved at the speed of a table system.



(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平4-213129

(43) 公開日 平成4年(1992)8月4日

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/02	5 0 1	8841-5B		

審査請求 未請求 請求項の数2(全 6 頁)

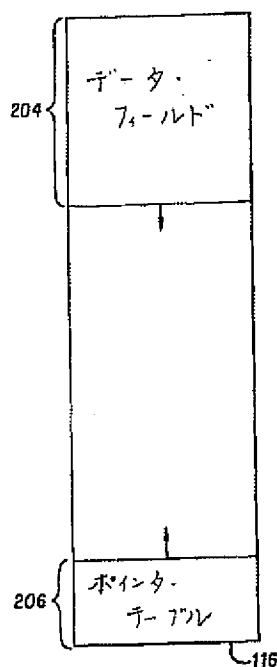
(21) 出願番号	特願平3-31890	(71) 出願人	590000400 ヒューレット・パツカード・カンパニー アメリカ合衆国カリフォルニア州バロアル ト ハノーバー・ストリート 3000
(22) 出願日	平成3年(1991)1月31日	(72) 発明者	フィリップ・ガルシア アメリカ合衆国カリフォルニア州サラト ガ、ダンディー・アベニュー 18872
(31) 優先権主張番号	4 7 2 0 5 5	(74) 代理人	弁理士 長谷川 次男
(32) 優先日	1990年1月31日		
(33) 優先権主張国	米国 (U S)		

(54) 【発明の名称】 メモリ管理システム及びメモリ管理方法

(57) 【要約】

【目的】 データ・メモリへのデータ格納効率を向上し、かつ検索速度も速く保つ。

【構成】 データ・メモリの1端よりポインタ・テーブルを開始成長させ、他端よりデータ・フィールドを開始成長させる。連結リスト方式の格納効率とポインタ・テーブル方式の検索速度が得られる。



(2)

特開平4-213129

1

【特許請求の範囲】

【請求項1】後記(イ)及至(ニ)より成り、後記(ホ)の特徴を有するマイクロコンピュータにおけるメモリ管理システム。

(イ)第1端と第2端を有するデータ・メモリ領域。

(ロ)1つあるいは複数のデータ・ポインタを有する可変長のポインタ・テーブルを動的に割当ててするための第1手段。前記データ・ポインタは前記第1端より開始する。

(ハ)データ・フィールド内に可変長のデータ・ブロックを動的に割当ててために、次の使用可能メモリ位置を前記ポインタ・テーブルから獲得するための第2手段。前記データ・フィールドは前記データ・メモリ領域の前記第2端から開始する。

(ニ)前記1つあるいは複数のデータ・ポインタに前記データ・メモリ領域の前記データ・フィールドに関する更新された位置情報を設定するための第3手段。

(ホ)前記ポインタ・テーブルと前記データ・フィールドのそれぞれは、それぞれ前記データ・メモリ領域の前記第1端と前記第2端のそれぞれから互いに成長する。

【請求項2】後記(イ)及至(ニ)のステップから成るマイクロコンピュータにおけるメモリ管理方法。

(イ)データ・メモリ領域の第1端から開始するデータ・ポインタを含む可変長ポインタ・テーブルを動的に割当てるステップ。

(ロ)前記ポインタ・テーブルから次に使用可能なデータ・メモリ位置のアドレスを獲得するステップ。前記使用可能データ・メモリ位置は前記データ・メモリ領域の第2端から開始し、データ・ブロックが書き込まれる。

(ハ)前記次に使用可能なデータ・メモリ位置に前記データ・ブロックを書き込むステップ

(ニ)前記ポインタ・テーブルに後続データ・ブロックの位置情報を含むポインタを割当ててステップ。前記後続データ・ブロックは前記データ・メモリ領域と連続している。

【発明の詳細な説明】

【0001】

【発明の技術分野】本発明は、マイクロコンピュータにおいてメモリの効率的な割当て及びアクセスを行なうためのシステム及び方法に関する。

【0002】

【従来技術とその問題点】最新のコンピュータ・システムの場合、データ記憶の管理方法が、コンピュータの効率的な操作において重要な役割を果たしている。コンピュータ・システムは、複雑さを増しているため、メモリ管理方式が、しだいに重要になっている。利用可能なメモリの効率的利用だけでなく、データに対するアクセス速度も、こうした管理方式を開発する場合に考慮すべき重要な要素になっている。

【0003】個々のデータ項目の数及びサイズが未知と

2

いう状況の場合、融通性のあるメモリ管理方式によって、コンピュータ・システムが貴重なメモリを最も賢明に利用することが可能になる。これは、特に、データ集録を伴う方式の場合にあてはまることである。こうした方式の一例が、店で行なわれる購買を記録することが可能なコンピュータ・プログラムである。ユーザは、購入される各品目毎に、品目名、顧客名、さらには、(書き入れられる場合もあれば、そうでない場合もあるが)クレジットカードの番号のようなオプション項目の数字といったデータをプログラムに入力することができる。データ項目のサイズ及び数は、前もって決めることができないという点に留意すべきである。データの入力前に、潜在的項目毎に多量のメモリを割り当てるのは無駄であり、このオプション・フィールドを利用しない場合、メモリの割当ては全くの無駄になる。データの入力がすむと、データの検索をすばやく行なえることが、やはり、極めて重要なことになる。

【0004】このメモリ管理問題の解決に当たって、メモリを効率よく管理するため、いくつかの異なる方式が工夫されてきた。テーブルの利用を伴う方式と、連結リストを伴う方式という、2つの一般的な方式が存在する。

【0005】テーブルによるメモリ管理方式の場合、ポインタ・テーブル及びデータ領域とすべき、2つの固定領域の割当てが必要になる。データの記憶前に、ポインタ・テーブル及びデータ領域に利用するため、一定量のメモリがメモリ管理ルーチンによって割り当てられる。データ記憶のためにメモリが必要になると、データは、メモリの利用できる次のブロックに書き込まれ、ポインタ・テーブル内のポインタに、このデータ・ブロックのアドレスが与えられる。次に、後続のデータ・ブロックとすべき部分のアドレスが計算され、保持される。特定のデータ・ブロックが要求されると、要求されたデータ・ブロックのアドレスを見つけるには、ポインタ・テーブルを探索するだけですむ。ただし、テーブルの利用には、データの記憶前に、ポインタ・テーブルとデータ領域の割当てを行なわなければならないという問題がある。従って、一定数のポインタと、データ領域のための一定数のスペースを割り当てなければならない。これは、例えば、100のポインタに対するスペースを割り当てておいて、100を超えるデータ・ブロックの記憶が必要であるということが判明した場合、ポインタ数は不十分ということになる。また、少数の極めて大きいデータ・ブロックにしか割り当てられない場合も、問題になる。その場合、少数のポインタしか利用されないため、残りの未使用のポインタが記憶されているメモリが、無駄になる。

【0006】テーブルよりもメモリ割当ての効率のよいもう1つの一般的なメモリ割当て方式は、連結リストとして知られている。この方式の場合、各データ・ブロッ

(3)

特開平4-213129

3

ク毎にポインタを保持するためのテーブルを設けるのではなく、割り当てられた各データ・ブロックの指定部分が、利用される次のデータ・ブロックに対するポインタとして用いられる。従って、新しいデータ・ブロックの生成時にのみ、ポインタの割当てしか行なわれない。この方法論の一例として、割り当てられたデータ・ブロックが2つ存在する場合、第1のブロックのポインタが、割り当てられる第2のブロックを示す。第2のブロックが割り当てべき最後のデータ・ブロックである場合、該データ・ブロックのポインタは、利用できる次のデータ記憶場所を示す。こうして、所望のブロックが得られるまで、ソフトウェア・プログラムは、データ・ブロックのリンクをたどることができる。

【0007】ただし、連結リスト方式には、所望のデータ・ブロックを見つけるのに長時間を要するという欠点がある。例えば、所望のブロックが、長い連結リストをなすデータ・ブロックの終端に位置する場合、そのデータ・ブロックを探索するソフトウェア適用業務は、所望のデータを得る前に、連結リストの全長にわたって探索しなければならない。

【0008】要するに、この一般的な従来技術のメモリの割当て方式（テーブル及び連結リスト方式）は、両方とも、欠点を有していることになる。テーブルは、ポインタを必要とするので、メモリのスペースが無駄になり、一方、連結リストは、メモリの効率は高いが、データ検索速度の低下する場合がよくある。従って、明らかになったことは、連結リストの場合と同様に効率よくメモリの割当てを行なうことができ（スペースに関して）、また、テーブルの場合と同様に、所望のデータ・ブロックをすばやく見つけ出す能力も備えたメモリ管理方式が必要という点である。

【0009】

【発明の目的】本発明の目的は、連結リスト方式の効率でメモリの割当てを行ない、テーブル方式の速度でデータの検索を行なう、マイクロコンピュータ上のメモリの管理システム及び方法を提供することである。

【0010】

【発明の概要】本発明の実施例においては、ポインタ・テーブルは、所定のデータ記憶領域の一方の端から始めて、動的に割当てが行なわれる。このポインタ・テーブル内にあるポインタは、それぞれ、割り当てられたデータ・ブロック、または、利用可能な次のデータ記憶場所を示している。利用可能な次のデータ記憶場所を示すポインタは、当初、ポインタ・テーブルの割当てを始めたデータ記憶領域のもう一方の端を示す。そのデータ記憶領域にデータ・ブロックが書き込まれるにつれて、ポインタ・テーブル及びデータ・ブロックが、互いに対して増大し、中断しなければ、その増大が継続することによって互いに重なり合うまで、その状態を続けることができる。

4

【0011】データ記憶領域に対するデータ・ブロックの書き込みが要求されると、本発明では、利用可能な次のデータ記憶場所を示すポインタをポインタ・テーブルから入手して、チェックを行ない、データ・ブロックの書き込みによって、データ・ブロックとポインタ・テーブルが重なり合わないことを確認する。重ならない場合、データ・ブロックの書き込みが行なわれる。その後、次のデータ・ブロックを書き込むべき場所を含む新しいポインタが割り当てられる。この場所は、利用可能な次のデータ記憶場所であり、前に書き込まれたデータ・ブロックに隣接している。

【0012】

【望ましい実施例の詳細説明】本発明は、マイクロコンピュータにおける効率的なメモリの割当て及びアクセスに関するシステム及び方法である。すなわち、本発明は、メモリの効率のよい割当てを可能にし、かつ、記憶されているデータの迅速な検索が行なえるようにすることによって、マイクロコンピュータ100（図2に示す）またはコントローラにおけるメモリの管理を効率よく行なうためのメモリ割当てルーチンを与える。

【0013】望ましい実施例の場合、本発明は、図2に示すコンピュータ環境において実行されるのが普通である。図2から分るように、本発明（実施例の1つでは、メモリ割当てルーチン114）は、より大規模なコンピュータ・プログラム112の一部をなすのが望ましい。このコンピュータ・プログラム112は、標準的なソフトウェア適用業務の場合もあれば、マイクロコンピュータ100上のオペレーティング・システムまたはB I O Sの場合もある。

【0014】メモリ割当てルーチン114とそのコンピュータ環境との対話の方法については、下記の例によって最もよく理解することができる。ユーザがコンピュータ・プログラム112と対話中の場合、ユーザは、入力装置102によってデータを入力する。次に、データは、バッファ104に送られて、そこに記憶される。中央演算処理装置（CPU）108にバッファに読み取るべきデータがあることが伝えられ、このデータが読み取られる。

【0015】メモリ割当てルーチン114は、データ記憶領域116内におけるスペースの割当てを行い、CPU108に対し、バス106を介してデータ記憶領域116にデータを送るように命令する。このようにメモリ割当てルーチン114は、実際には、入力装置102を介したユーザによるデータ入力によって開始される。もちろん、自動的にデータを生成し、次に、メモリ割当てルーチン114によって記憶されるように、コンピュータ・プログラム112を書くことも可能であり、あるいは、以上の何らかの組合せも可能である。

【0016】本発明は、サイズの異なるデータ・ブロックがどれだけ要求されることになるのか分からなく、ま

(4)

特開平4-213129

5

た、データに対するアクセス速度が重要になるコンピュータ・システムで実施されることを意図したものである。従って、次に、図1に関連して本発明の説明を行なうことにする。

【0017】図2には、データの記憶に用いられるべきデータ記憶領域116の概要が示されている。このデータ記憶領域116は、マイクロコンピュータ100（図2に示す）内におけるフリー・メモリ全体の場合もあれば、そのより小さい部分の場合もある。実際、いくつかのデータ記憶領域116を1度にセット・アップして、利用することも可能である。一般に、ポインタ・テーブル206の割当ては、メモリ割当てルーチン114によって、データ記憶領域116の一方の端から開始される。このポインタ・テーブル206には、データ・フィールド204内におけるデータ・ブロックを示すポインタが含まれる。これらのデータ・ブロックの割当ては、ポインタ・テーブルによるデータ記憶領域116のもう一方の端から開始される。データ・ブロックが記憶されているデータ記憶領域116の可変長部分は、本書においてデータ・フィールド204と呼ばれる。従って、新しい項目が追加されるにつれて、データ・フィールド204及びポインタ・テーブル206が、互いに増大するのは明らかになる。

【0018】メモリ割当てルーチン114によって、新しいデータ・ブロックがデータ・フィールド204内に割り当てられる毎に、ポインタ・テーブル206に新しいポインタも割り当てられる。この新しいポインタの内容は、利用可能なメモリ内において、次のデータ・ブロックが格納される位置の開始アドレスを示している（すなわち、それは、利用可能な次のデータ記憶場所を示している）。従って、ポインタ・テーブル206の各ポインタ及び各データ・ブロックは、動的に割り当てられる。従って、前もって、ポインタ・テーブル206またはデータ・フィールド204に関して明確にメモリの割当てを行なう必要はない。

【0019】上述のメモリ管理方式は、入力方法に関係なく、データを効率的に扱うことができるので、大幅なフレキシビリティが得られる。例えば、全てのデータ・ブロックが最小サイズになるが、その数は極めて多いという場合がある。これによって、各データ・ブロック毎に1つのポインタが生成される。こうした場合、本発明では、データ記憶領域116が完全に満たされるまで、ポインタを割り当てることが可能である（すなわち、このようにしなければ、ポインタ・テーブルは、データ・フィールドと重なり合う）。実際に、データ・ブロックがデータを含んでいない（後述のように空である）可能性もあるので、ポインタ・テーブルは、データ記憶領域116を有効に使い切ることが可能である。

【0020】同様に、データ記憶領域116のほぼ全てを占めるデータ・ブロックが1つだけしか割り当てられ

6

ない場合にも、これは可能である。両方の場合とも、メモリが浪費されることはなく、どのデータ・ブロックに対する探索であっても、すぐに見つけだすことができる。従って、本発明は、標準的なテーブル方式と同じ速さでデータ・ブロックを突きとめることが可能であり、連結リスト方式と同様にスペース効率がよい。

【0021】次に、図3に関連し、本発明に関するさらに詳細な説明を行なうことにする。本発明では、データ・ブロックの割当て前に、まず、フリーなポインタ索引に含まれているメモリ・アドレスを読み取る。このメモリ・アドレスは、用いられる次に利用可能なポインタを示している（これは、前のデータ・ブロック割当て時にセット・アップ済みである）。次に、本発明は、図3において“ポインタn”として示された、この次に利用可能なポインタの内容を読み取る。ポインタnには、次のフリーなデータ・ブロック（データ・ブロックn）のアドレスが含まれている。次に、チェックを行なって、新しいデータ・ブロック及び新しいポインタを割り当てることによって、データ・フィールド204がポインタ・テーブル206と重なることにならないか確認する。新しいデータ・ブロック及びポインタの生成のために十分なデータ記憶領域116が残されていないければ、エラー・メッセージが生じる。十分な領域が残されていれば、データ・ブロックnで始まるメモリ・アドレスを備えたデータ・ブロックが、生成される。

【0022】データ・ブロックの書き込みが完了すると、後続のデータ・ブロック（データ・ブロックn+1として示されている）の開始位置のアドレスが得られる。このアドレスは、さらに、利用可能な次のポインタ位置（ポインタn+1）に納められ、フリーなポインタの索引が、ポインタn+1を示すように更新される。データ・ブロックが空のデータ・ブロックの場合（後述する状況）、次のデータ・ブロックの開始アドレスを更新する必要はない。

【0023】次に、その特定の利用例に関連し、本発明の望ましい実施例について説明する。従来、マイクロコンピュータにおける構成設定は、マイクロコンピュータ内のトグル・スイッチによって手動でしなければならなかった。別の構成設定は、マイクロコンピュータの一般的な記憶スペース内に構成ファイルをセット・アップすることによって、セットされる。マイクロコンピュータは、一般に、ユーザによって1度だけセット・アップされる不揮発構成の情報記憶領域を含むように設計されているが、最近のコンピュータは、この概念をより十分に利用している。こうした記憶領域内の情報例には、マイクロコンピュータのスロット内のハードウェア（すなわち、図4に示す背面412に取りつけられる）に関するタイプや、シャドーイングを施すのが予約メモリのどの部分かといったものがある。

【0024】次に、第4図に関連し、とりわけ、背面4

(5)

特開平4-213129

7

12に取りつけられるハードウェアの構成情報に関する上述の本発明の利用について説明する。図4を参照すると、114で示された本発明のルーチンは、マイクロコンピュータ100の基本的入力/出力システム(BIOS)402の一部である。背面412のスロット内にあるハードウェア装置の1つ(本書ではカード408と呼ぶ)が、背面412に加えられる時、ユーザは、構成ユーティリティを実行する必要がある。この結果、メモリ割当てルーチン114によって、背面412に納められるカードのタイプを表わした構成情報がCMOSメモリ406に書き込まれる。この構成情報は、上述の案に基づき、メモリ割当てルーチン114によってメモリに書き込まれる。空スロット418が、空の状態のままである場合、本実施例では、後で情報が生じると、一種の位置マーカとして、それが構成情報の納まる位置であることを表わしたポイントを備える空のブロックが、生成さ

8

れる。もちろん、本発明は、構成情報環境におけるメモリ管理に対する利用に限定されるものでないことは明らかである。

【0025】

【発明の効果】以上詳述したように、本発明の実施により、従来技術の連結リスト方式と同等の効率でデータの格納が可能となり、かつ、テーブル方式と同等の検索速度が得られると言う実用上有益な効果が得られる。

【図面の簡単な説明】

【図1】本発明の一実施例によって生成されるメモリ構成の上位レベルの概略図である。

【図2】本発明を実施する環境の上位レベルの概略図である。

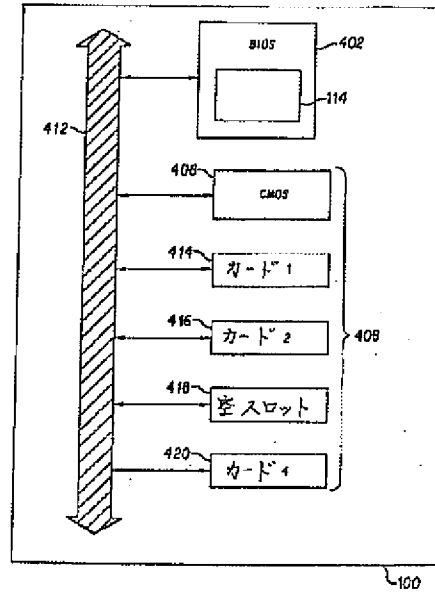
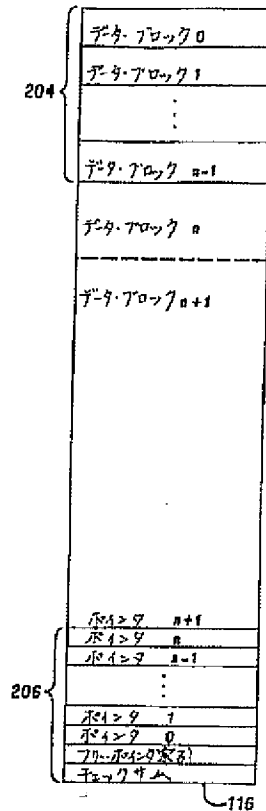
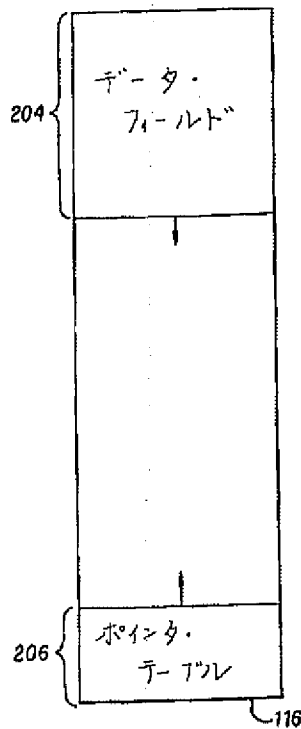
【図3】本発明の一実施例によって生成されるメモリ構成の下位レベルの概略図である。

【図4】本発明の一実施例を説明するための図である。

【図1】

【図3】

【図4】



(6)

特開平4-213129

【図2】

